Requirements Engineering 1: "Introduction"

Steve Easterbrook

8/20/97

Specification

It is necessary to transport an egg over a distance of at least 1 metre without direct intervention. The egg must not be broken or cracked. The egg must not make contact with the ground. No person is allowed within 1 metre of the stopping point of the egg.

Source: *Based on an exercise devised by Prof Anthony Finkelstein*

Egg Race Lessons

- **Whenever you write a spec, someone will have to implement it**
- **You cannot interpret a spec unless you understand the customer's need**
- **Beware of cultural words/phrases**
- **Systems are often evaluated against non-functional criteria**
- **Unless the spec is explicit about how to measure success, your assumptions will be wrong**
- **Software Engineering is about building things...**

Software Requirements Engineering

Elicitation
Modeling
Analysis
Specification
Verification (and Validation)
Requirements Management

Software Requirements vs System Requirements

- **Software is always a component of a larger system**
- **If the system already exists:**
  - **software requirements are derived from users needs and capabilities of the hardware**
  - ***"application engineering"***
- **If the system is being developed concurrently:**
  - **software requirements are derived from the system specification**
  - **system and software requirements must co-evolve**
  - **software is always regarded as mutable!**
  - ***"systems engineering"***

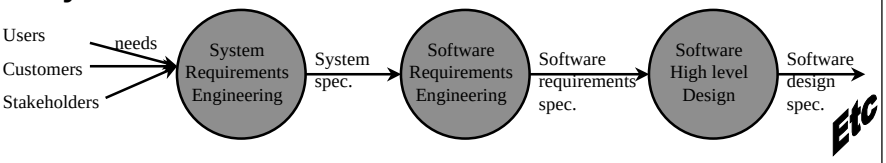
System Requirements Engineering

- **Transforms operational need into:**
  - **system description**
  - **system performance parameters**
  - **system capabilities**
- **The process involves analysis, design, trade-off studies, prototyping, etc.**
- **System description defines an architecture:**
  - **functional requirements are allocated to the various subsystems**

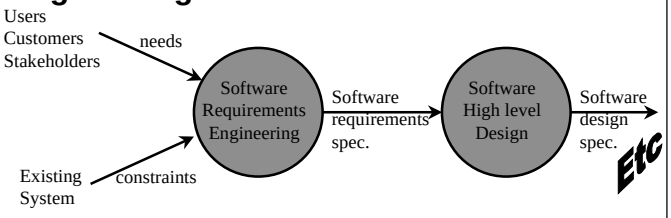
Source: *Adapted from Thayer & Dorfman, 1997, p41*

Types of software requirements engineering

- **Systems Software**



- **Application Engineering**



Why worry about requirements?

- **Engineers find it hard to write good specifications**
- **Managers tend to truncate the requirements process ("it's unproductive"!!)**
- **Customers often can't or won't validate requirements**
- **Huge range of choice for representations & methods**
- **Gap between system requirements and software requirements**
- **System engineers don't have the knowledge or skill for allocation of system requirements to software**
- **Requirements always change as a system is developed**
- **A system cannot be tested unless the requirements are understood**

Source: *Adapted from Thayer & Dorfman, 1997, p3*

Next Week...

- **Requirements Engineering in the development lifecycle**
  - **Why requirements errors are so expensive**
  - **The requirements process**
- **Debate:**
  - **"This house believes that it is not necessary to specify requirements before building a software system"**

References

- Thayer, R. H and Dorfman, M. (eds.) "Software Requirements Engineering, Second Edition". IEEE Computer Society Press, 1997.